POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Analysis of Milan Personal Services - Database 12

Group Project Report in DATA INFORMATION AND QUALITY (2024-2025)

Mauro Orazio Drago, 10947649
Dennis Pierantozzi, 10959535
Davide Morelli, 10962902

1. Introduction

The aim of this report is to present our project for the 'Data and Information Quality' exam. The project requested to execute a complete Data Preparation Pipeline on the dataset that has been assigned to our group following precise steps. Specifically, we conducted:

- Data Profiling and Assessment
- Data cleaning
- Data Analysis.

To support our work we created also a GitHub repository to store all notebooks and datasets.
The repository can be found at: `https://github.com/madratak/Milan_Services_2012`

1.1. Set up

The dataset pertains to beauty services in Milan and is openly available at DatiOpen.it. It contains various details about each shop, including address, type of exercise (e.g., hairdresser, barber, beautician), main activity, and area in square meters.
We used Pandas, NumPy, Sklearn, Rapidfuzz and Scipy libraries for processing and Kaggle as our development environment. To maintain a clear structure, we divided our code into three notebooks, which are reflected in the section of this report.
External knowledge was essential for certain aspects of our assessments and cleaning process, so we utilized a set of datasets sourced from Geoportale.comune.milano.it. These datasets included a comprehensive list of all streets and street types in Milan, as well as a catalog of address numbers for each street in the city. In our report, these external datasets will be referred to as 'VIARIO' or 'STRADARIO.'

2. Pipeline Implementation

2.1. Data Assessment

The dataset consists of 3909 rows and 10 columns:

- 'Tipo esercizio pa': type of work performed (e.g., hairdresser, beautician)
- 'Ubicazione': full address (type of public space, name of the public space, address number, municipality number)
- 'Tipo via': type of public space
- 'Via': name of the public space
- 'Civico': address number
- 'Codice via': unique identifier of the public space
- 'Zd': municipality number of Milan
- 'Prevalente': main activity carried out
- 'Superficie altri usi': square meteres not used directly for the business
- 'Supericie lavorativa': square meteres used directly for the business

2.1.1 Data Types and Initial Observations:

- Most columns are of type 'object,' except for 'Codice via,' 'Superficie altri usi,' and 'Superficie lavorativa,' which are float64.
- The dataset includes one duplicate row.

2.1.2 Data Quality Dimensions:

Data Assessment aims to define a score for the data quality dimensions chosen. This can be done in a subjective or objective way. The subjective method requires the usage of questionnaires to assess the scores through the answers gathered. This process requests also a final interpretation step.
The objective way, instead, measures the scores using mathematical tools and formulas. Because of temporal limits, we have decide to perform an objective assessment only.

We assessed completeness, consistency, and accuracy but excluded timeliness due to the absence of temporal metadata.

Completeness
Completeness is the degree to which a given data collection includes the data describing the corresponding set of real-world objects. Ojbectively the completeness is computed as the total number of not null values divided by the total number of cells.
The completeness of our dataset is 79%.
The columns that contains the major number of null values are 'Prevalente' and 'Superficie altri usi'.

Consistency
Consistency is the satisfaction of semantic rules defined over a set of data items. Considering our dataset we have defined some specific rules that we have checked.

- Rule 1: The 'Ubicazione' column must equal the concatenation of 'Tipo via,' 'Via,' 'Civico,' and 'Zd.'
  Issues found:
  - 152 mismatches with 'Civico'
  - 21 mismatches with 'Tipo via'
  - 88 mismatches with 'Via'
  - 18 mismatches with 'Zd'
- Rule 2: Values in 'Superficie lavorativa' and 'Superficie altri usi' must be non-negative.
  Result: no negative values were found.

Accuracy

Accuracy dimension check the extent to which data are correct, reliable and certified. There are two types of accuracy: the syntactic and semantic accuracy. The former checks whether or not a value is correct syntactically with respect to a domain of allowed values. Semantic accuracy checks whether or not a value has a real meaning in the real world.

- **Semantic Accuracy**: We have verified the validity of 'Tipo via' and 'Via' against external datasets (TIPOVIA and VIARIO).
  - 'Tipo via': 99.9% accuracy (3905 correct out of 3908 non-null values).
  - 'Via': 97.2% accuracy (3800 correct out of 3908 non-null values).
- **Syntathic Accuracy**: We have checked if the values in column 'Zd' are in the number of municipality of Milan (1-9).
  - 'Zd': 99.97% accuracy (2 invalid values out of 3909).

2.2. Data Profiling

Data Profiling is the set of activities and processes designed to determine the metadata of a given dataset.

We have performed a single column analysis for each column that has discrete values, inspecting the number of unique values for each of them and counting the most and least common ones while for the continuous columns 'Superficie altri usi' and 'Superficie lavorativa' we have studied the value distributions.

Single Column Analysis

- 'Tipo esercizio pa': 103 unique values (most frequent: 'Parrucchiere per signora,' least frequent: 'Truccatore').
- 'Ubicazione': 3554 unique values.
- 'Tipo via': 17 unique values (most frequent: 'VIA,' least frequent: 'SIT').
- 'Via': 1370 unique values.
- 'Civico': 235 unique values.
- 'Codice via': 1376 unique values.
- 'Zd': 10 unique values (1-9).
- 'Prevalente': 63 unique values.

Value distribution

- 'Superficie altri usi': Mean = 9.56, Std Dev = 13.73
- 'Superficie lavorativa': Mean = 40.15, Std Dev = 27.73

Below are reported two visualization (e.g., histograms and boxplots). They demonstrate positive skewness for both fields, suggesting that median imputation is preferable for missing values.

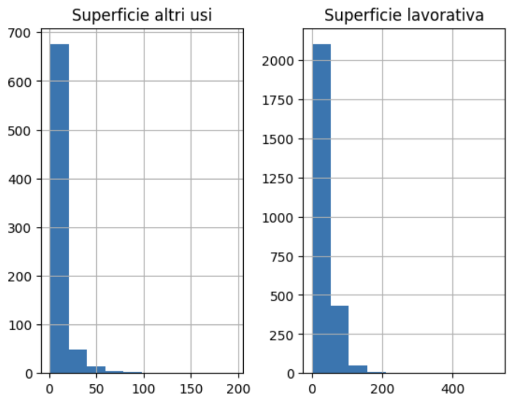

Figure 1: Histograms with the distribution of 'Superficie altri usi' and 'Superficie lavorativa'

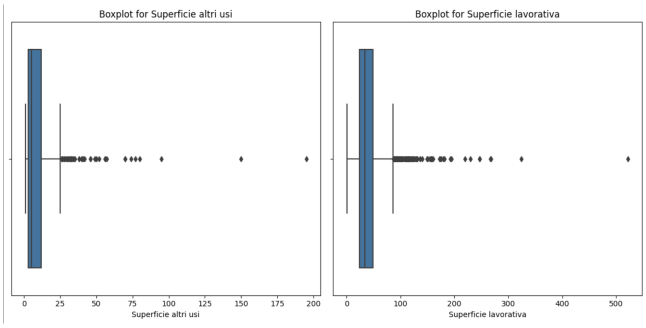

Figure 2: Boxplots for the distributions of 'Superficie altri usi' and 'Superficie lavorativa'

2.3. Data Cleaning

Data cleaning is the process of identifying and eliminating inconsistencies, discrepancies and errors in data in order to improve quality.

2.3.1 Data Wrangling

Column Renaming

To improve readability and data management, the dataset columns were renamed using a standardized format: lowercase names separated by underscores. This facilitates access and manipulation of the data during subsequent analysis phases. Specifically:

- Tipo esercizio pa → t_es (business type)
- Ubicazione → ubic (location)
- Tipo via → t_via (street type)
- Via → via (street name)
- Civico → civ (street number)
- Codice via → codice_via (street code)
- ZD → zd (zone designation)
- Prevalente → main (primary activity)
- Superficie altri usi → sup_alt (surface area for other uses)
- Superficie lavorativa → sup_lav (working surface area)

2.3.2 Data Transformation

Data transformation involved several operations to standardize and enrich the dataset, aiming to prepare it for more in-depth analysis.

Correction of 'zd' values:
Inconsistent values in the 'zd' column were identified and corrected. In particular, rows representing the same object were merged, ensuring the consistency of 'zd' values.

Reduction of labels in 't_es':
The 't_es' column contains 103 unique labels. Many of these labels have similar meanings or represent variations of the same concept, leading to redundancy and potential inconsistencies in analysis. To address this, we have reduced the number of labels by grouping similar ones under a unified label. This will make the dataset easier to interpret and analyze while preserving the essential distinctions between categories.

This transformation process primarily involved two key phases:

1. *Composite Label Handling:* Certain rows included labels such as 'Tipo A-B-C-D' or 'Parrucchiere misto,' which required splitting or expanding into multiple categories to accurately reflect all included elements;
2. *Label Mapping:* After handling composite labels (e.g., splitting ''TIPO A-B-C-D'' into individual components), the code replaces each original label (such as ''Centro abbronzatura'') with the most similar label from a predefined set using the Jaccard similarity measure. This step ensures that similar labels are grouped under a single, consistent label based on the highest similarity score.
   The predefined set allowed us to overcome the issue of syntactic differences between labels. For example, the key 'Tipo A - Estetica Manuale' is matched with the set ['Tipo a - estetica manuale', 'Estetista', 'Estetica', 'Manicure', 'Pedicure', 'Unghie']. This ensures that labels like 'Pedicure', for instance, are matched correctly. The similarity is measured for each label in the set, and the highest value is chosen. The values in the set were selected by manually reviewing the data and grouping labels that share the same semantic meaning.
   Moreover we have reviewed the regulations governing personal services in Italy, which categorize beauty services into types A, B, C, and D, each representing broad categories encompassing various activity types. These distinctions were maintained during the cleaning phase. The 'Acconciatore' and 'Parrucchiere' labels were merged into one category in accordance with a 2018 law. However, since our dataset spans from 2012, we chose to preserve this distinction within the data.

After all the mappings we ended up with:

- 907 Acconciatore
- 601 Parrucchiere per Uomo
- 1294 Parrucchiere per Donna
- 1266 Tipo A - Estetica Manuale
- 693 Tipo B - Centro di Abbronzatura
- 226 Tipo C - Trattamenti Estetici Dimagrimento
- 162 Tipo D - Estetica Apparati Elettromeccanici
- 29 Centro Benessere
- 139 Centro Massaggi
- 24 Esecuzione di Tatuaggi e Piercing
- 1 Truccatore

Reduction of labels in 'main':
The 'main' column contains numerous null values, and the few non-null entries it holds offer no substantial difference from the data already present in the 't_es' column.
We decided to repurposed this column to complement the values in 't_es' when they were missing, thereby consolidating all relevant information into a single, unified column.
In order to do that we first complete a mapping as done in the previous step for the column 't_es' using the predefined set and the Jaccard similarity. Then we move the values from the 'main' column to the 't_es' column. After performing this merging with column 't_es' we dropped the main column

as it contained no more useful information fo us.

Format Verifications:

- *t_via*: contained an invalid type 'Vie'. We have mapped this value to the type 'Via', which is one of the types allowed in our dataset TIPOVIA, that we have taken as ground truth.
- *cod_via*: all values were integers, so no changes were made.
- *civ*: We ensured that all values followed the expected format: 1-3 digits, optionally followed by an uppercase letter (e.g., 12A), a slash '/' with a single digit (e.g., 93/1), or a slash '/' and another series of digits and letters (e.g., 15/101).We utilized regular expressions to check the validity of the values in this column and invalid values were replaced with 'NaN', ensuring that only correctly formatted entries remained for further processing.
- When processing records, some entries in the via column contained redundant information that matched part of the 't_via' column. For example, a value for 'via' was CSO COMO, which included the redundant prefix CSO. To ensure consistency we removed the redundant portion. We then validated the values against those in the VIARIO dataset. Similarity scores were calculated between the via column and two descriptive fields, using the highest score to flag rows for review if they fell below a 75% similarity threshold. The similarity measure used in this case was the edit distance, implemented using the RapidFuzz library.

'Ubic' Structure and Format Checking:
We processed the ubic field to extract and organize its contents into distinct components: 'ubic_t_via' (street type), 'ubic_via' (street name), 'ubic_civ' (street number), 'ubic_zd' (zone description), and note (additional information).
Next, we standardized the values and validated them against the VIARIO dataset by calculating similarity scores between the ubic_via column and the DESCRITTIVO and DENOMINAZIONE columns of the external dataset. The highest similarity score was selected for each row, and rows with a similarity score below 75% were flagged for further review. For the rows that did not have a similarity score higher than 75%, we performed a second-level review using the Jaccard distance. With this similarity measure, most of the rows now exhibit a similarity of 50% or greater, indicating that their addresses match semantically. As a result, the ubic_via column can now be standardized to align with the format of the DESCRITTIVO column of the VIARIO dataset.

2.3.3 Error Detection & Correction

This section typically involves identifying and rectifying inaccuracies or inconsistencies within a dataset. In this section we aim to solve the mismatches between the 'ubic' columns ('ubic_t_via', 'ubic_via', 'ubic_civ', 'ubic_zd') and the corresponding main columns ('t_via', 'via', 'civ', 'zd'). Mismatches are identified by directly comparing values in these columns.
The mismatches we identified were:

- 20 for 't_via'
- 151 for 'civ'
- 76 for 'via'
- 17 for 'zd'

Subsequently, we compared both the 'ubic' column and the concatenation of the other address fields (t_via+via+civico+zd) with our VIARIO dataset to identify which records corresponded to actual streets. This comparison generated two new temporary flags for each row, indicating whether either of the two addresses was valid.
These flags were then used to determine which rows to modify, retain, or discard. The following scenarios emerged:

- *Both flags TRUE but with mismatched values:* In this case, the entire row was dropped, as the dataset provided valid but conflicting information, and we could not determine which address was correct.

- *Both flags FALSE:* These rows were also discarded, as they referenced incorrect information.
- *Rows with the ubic-flag TRUE and the other FALSE:* For these rows, the information in the 'ubic' column was transferred to the individual address fields.

Finally, we performed a consistency check to ensure that the concatenated values of 'tipo_via' and 'via' aligned with the encoded value in 'cod_via,' maintaining data consistency.
At the conclusion of this phase, we removed all columns related to 'ubic' and the flag columns, as they were no longer necessary. By this point, the dataset contained 3,845 rows.

2.3.4 Data Deduplication

The first stage of data deduplication focused on addressing non-exact matches. These situations occurred when records linked to the same address were split into multiple rows. The objective was to consolidate these rows into a single entry, where each unique address would include the concatenation of all associated t_es labels. For the 'sup_alt' and 'sup_lav' fields, we chose to retain the maximum value among the merging entries. This decision stemmed from the realization that simply summing the individual values would result in values that significantly deviated from the distribution of existing entries with similar labels. To further support this approach, we considered real-world scenarios where commercial establishments offering different services might share common spaces (such as desks, tills, bathrooms, entrances, and waiting rooms).
Redundant rows still present after this phase were dropped in order to keep only the merged ones.

Another refinement we implemented involved merging 'Acconciatore' with either 'Parrucchiere per Uomo' or 'Parrucchiere per Signora' whenever they appeared in the same row. We interpreted the latter two as specializations of the former, and therefore, in cases where they coexisted, we retained the more specific label. This decision was based on the belief that the more specialized label carried greater relevance for our analysis.
Lastly for this section, we removed rows with duplicate entries in the 't_es', 'cod_via', 'via' and 'civ' columns, and also discarded any rows with null values in 't_es' as they were irrelevant to our analysis. The 'note' column was also removed, as it no longer held any significant value.
At this stage, the dataset contained 3,523 records.

2.3.5 Outlier Detection & Correction

We began by filling in the missing values in both the 'sup_lav' and 'sup_alt' columns. Based on insights gathered during the profiling phase, we concluded that the median was the most appropriate value to fill the gaps in both fields, and thus we applied it.
In particular we have computed the median per each unique value of 't_es' and used it to filled the missing value per each row. This allow a coherent handling of missing values based on the type of business the row represent.
Next, we computed the z-score for both columns and identified as outliers any values deviating by more than 3 standard deviations from the mean.

- A total of 53 outliers were identified in the 'sup_lav' column.
- A total of 19 outliers were identified in the 'sup_alt' column.

These outliers were then removed.
At this stage, we had successfully obtained a fully complete dataset, free of any NaN entries.

2.3.6 Final Assessment

Once the cleaning phase was complete, the best way to quantify the improvements made to our dataset was to recompute the same assessments we performed initially and compare the results:

From this result, it is evident that most of the improvement came from Completeness, which was expected given the significant number of missing values in the initial dataset.

DQ-Measure	Old value[%]	New value[%]	Improvement[%]
Completeness	79.025	100	26.54
Accuracy on Tipo_via	99.923	100	0.08
Accuracy on Via	97.236	100	2.84
Accuracy on ZD	99.974	100	0.03

Table 1: Data Quality Measure Improvement Matrix

3. Result

3.1. Data Analysis

In this section, we performed data analysis on both the raw and cleaned datasets. We focused on a classification task for the 'Tipo esercizio pa' column, aiming to predict its value based on the other available data.

3.1.1 Dirty Dataset Pipeline

We selected only the columns likely to impact performance, discarding the others.
The features we retained are:

- 'Tipo esercizio pa'
- 'Tipo via'
- 'Via'
- 'Zd'
- 'Superficie altri usi'
- 'Superficie lavorativa'

Encoding
Some features were categorical, so we applied encoding methods. Since 'Tipo esercizio pa' had 103 unique values, we used Label Encoding with the `LabelEncoder` from the `sklearn` library. For the categorical columns 'Tipo via' and 'Zd,' we applied the one-hot encoding technique.

Handling Null Values
Rows with null values in 'Tipo esercizio pa' were dropped. For 'Superficie lavorativa' and 'Superficie altri usi,' missing values were filled with the median of the existing values computed on the values grouped by each 'Tipo esercizio pa' value.

3.1.2 Clean Dataset Pipeline

The cleaned dataset doesn't have any null values beacuase of the data preparation pipeline implemented.
The features we keep in this case were:

- 't_es'
- 't_via'
- 'zd'
- 'sup_alt'
- 'sup_lav'

Encoding
For the categorical features we have perfomed the same encoding used for the dirty dataset. We used Label Encoding with the `LabelEncoder` for the 't_es' values and one-hot encoding technique for the 't_via' and 'zd' values.

3.1.3 Performances

. We have dividied the datasets in training and testing (20%) and used two different classifiers for our task:

- Random Forest
- XGBoost

The results are displayed below:

Model-Dataset	**Accuracy**	**Precision**	**Recall**	**F1 Score**
RandomForest-DatasetDirty	0.196615	0.163688	0.196615	0.176864
RandomForest-DatasetCleaned	0.665685	0.652894	0.665685	0.656759
XGB-DatasetDirty	0.247396	0.208719	0.247396	0.211121
XGB-DatasetCleaned	0.756996	0.757881	0.756996	0.752541

Table 2: Model performance metrics for different datasets

From the results we can say that:

- The cleaned dataset significantly improves the performance of both models, particularly for XGBoost, which performs best when the data is well-prepared. The dirty dataset severely hampers model performance, especially for RandomForest.
- On both cleaned and dirty datasets, XGBoost outperforms RandomForest in terms of all metrics. This suggests that XGBoost is a more robust algorithm, especially when dealing with noisy or unclean data.